MACHINE LEARNING:
REPORT FOR PROJECT ON TWITTER SENTIMENT ANALYSIS

BY – PIYUSH, SRUJAN AND VIPUL

In this project we are given the dataset which contains 1600000 tweets along with the class each tweet belongs to. The classes are negative, positive or neutral. Our task is to train the machine learning model on this huge dataset to predict the class of new tweets, whether they are positive, negative or neutral.

So first we imported the dataset into colab file and converted to dataframe.

The dataframe looks as follows:

	polarity	id	date	query	user	text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....

These are the first five rows of dataframe.

Polarity shows the class of tweets and text shows the tweets.

0 – Negative , 2 – Neutral , 4- Positive

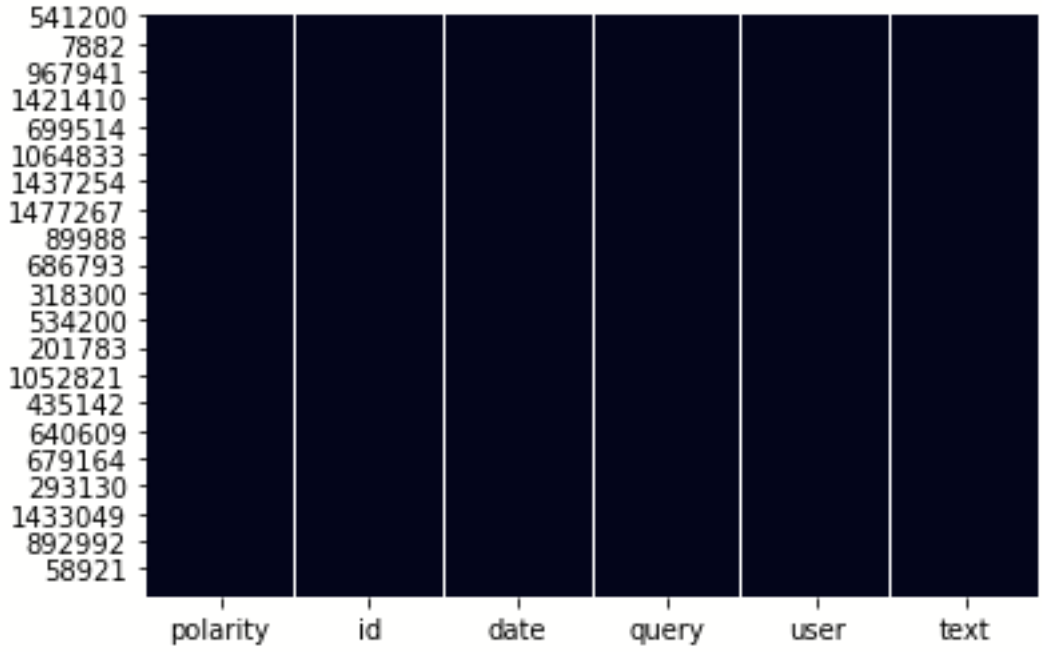
After doing some exploratory data analysis we found that , polarity contains only two classes which are 0 and 4. So there are no “Neutral” class tweets.

```
df['polarity'].unique()
array([0, 4])
```

Since there are two classes only , lets replace label 4 for positive class with 1.
So now labels are as follow :
0 – Negative , 1 – Positive .

Next we checked for NULL values in the dataframe:

```
sns.heatmap(df.isnull(), cbar=False)
```



So we can observe that dataframe do not contain any null values.

Some examples of tweets which belong to Positive class:

- **on lunch....dj should come eat with me**
- **@mrstessyman thank you glad you like it! There is a product review bit on the site Enjoy knitting it!**
- **@PerezHilton Zach makes me pee sitting down! And I'm a grown gay man!**
- **to sum up my day in one word ..... kackered!**
- **@k9wkj Great minds think alike**

So we can observe manually also that these tweets reflects positivity.

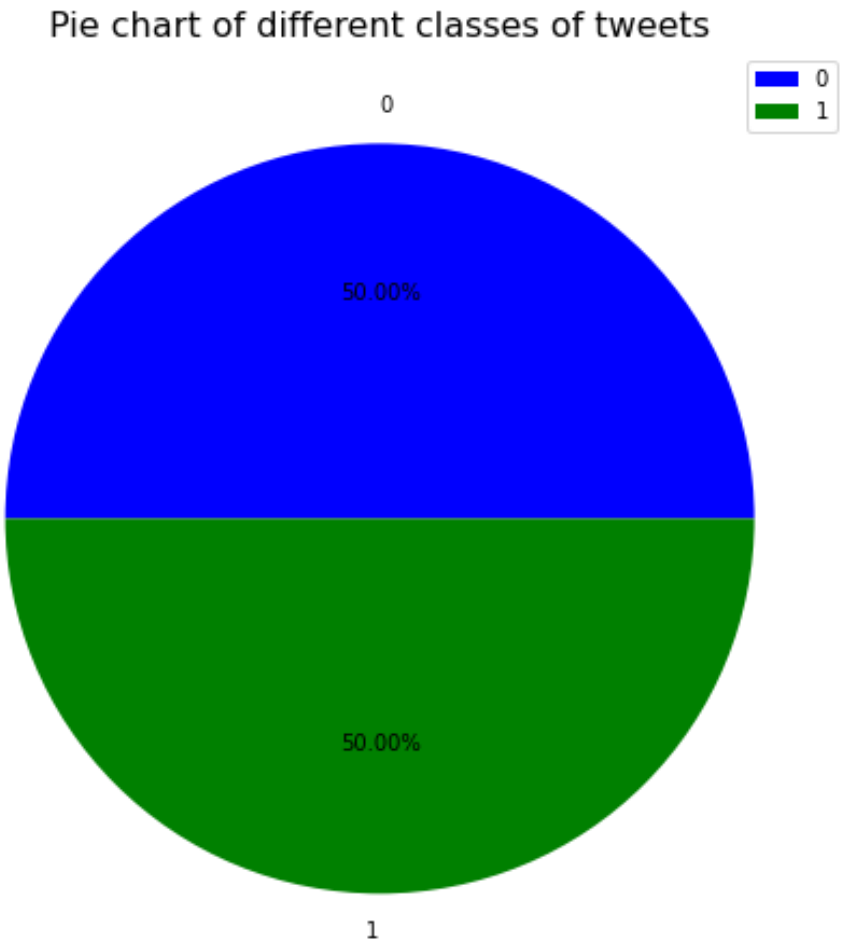
Some examples of tweets which belong to Negative class:

- **@chrishasboobs AHHH I HOPE YOUR OK!!!**
- **@misstoriblack cool , i have no tweet apps for my razr 2**
- **@TiannaChaos i know just family drama. its lame.hey next time u hang out with kim n u guys like have a sleepover or whatever, ill call u**
- **School email won't open and I have geography stuff on there to revise! *Stupid School* :'(**

- ☐ `upper airways problem`

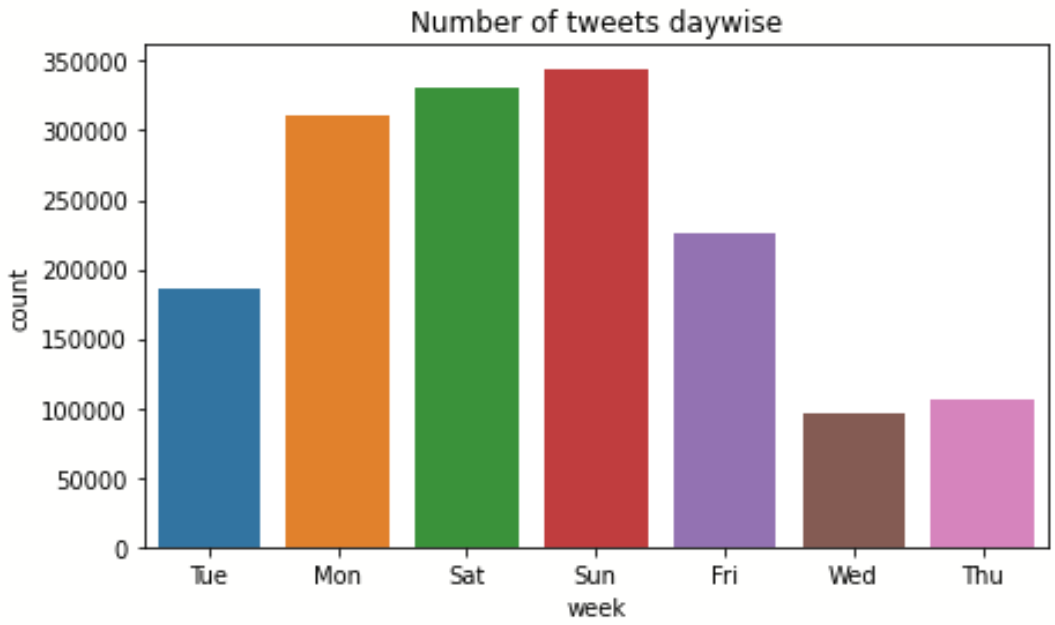
So we can observe that these tweets reflects negativity of some kind.

DISTRIBUTION OF TWEETS BY CLASSES:

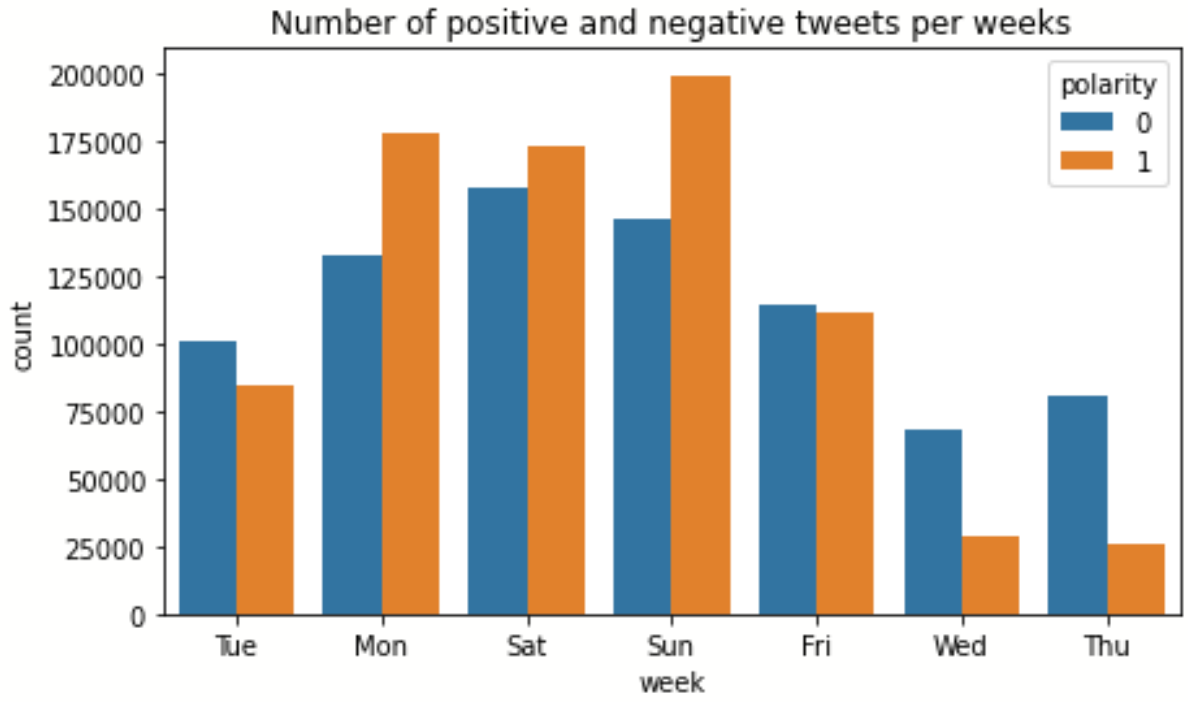


So we can see that there is an equal distribution of tweets in positive and negative class samples.

`Visualizing the number of tweets posted in days of a week`



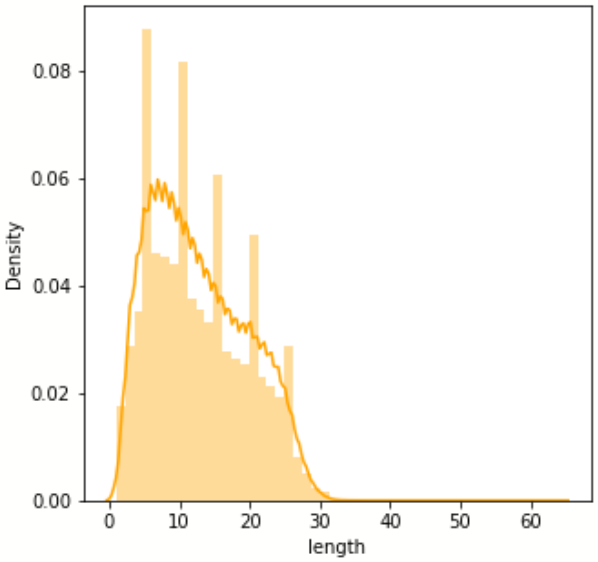
Visualizing the positive and negative tweets per weeks



Now let's see the distribution positive and negative tweets with their length:

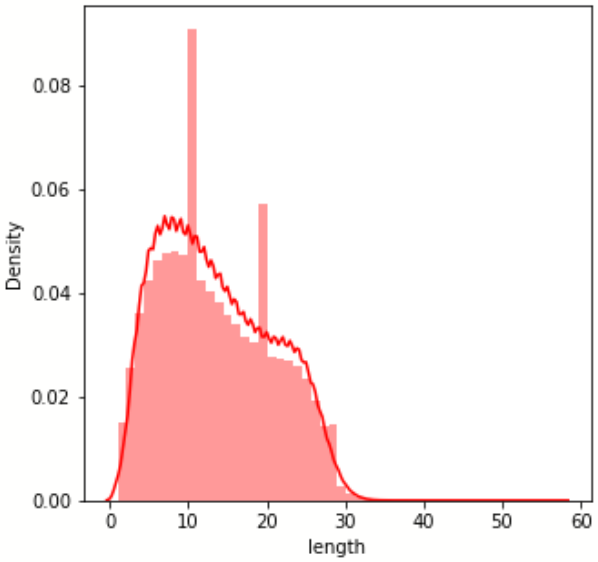
Distribution of text length for positive sentiment tweets.

	length
count	800000.0
mean	12.77
std	6.82
min	1.0
25%	7.0
50%	12.0
75%	18.0
max	64.0



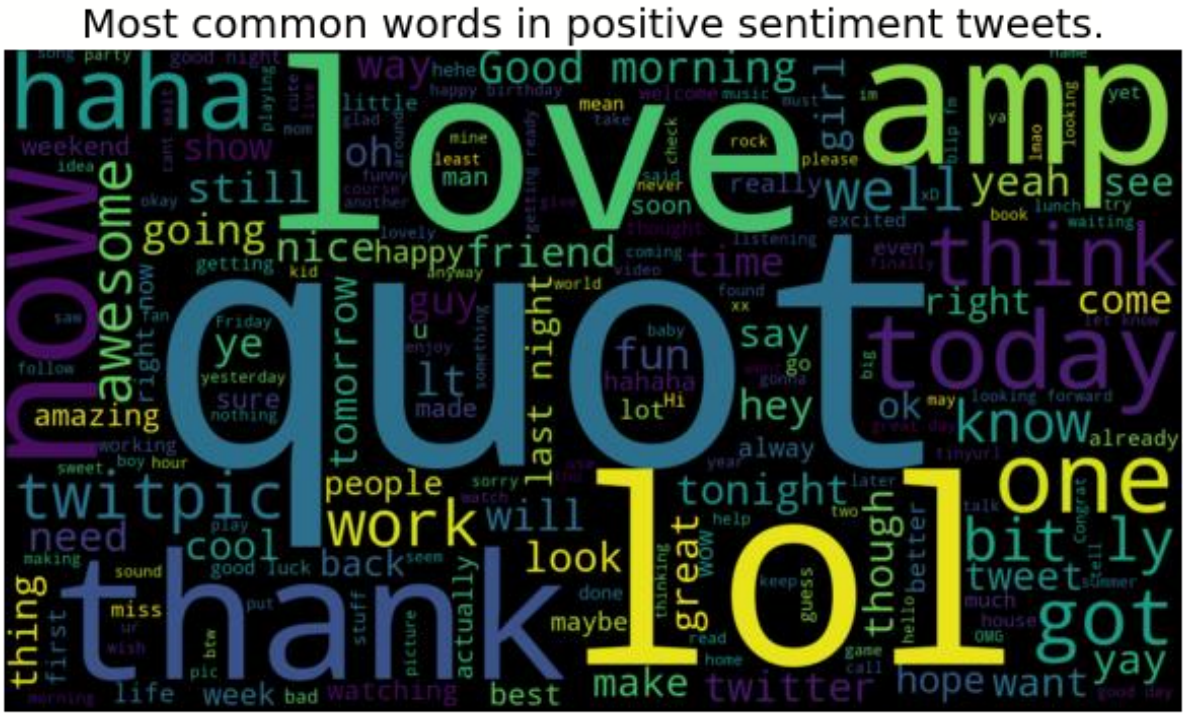
Distribution of text length for negative sentiment tweets.

	length
count	800000.0
mean	13.58
std	7.07
min	1.0
25%	8.0
50%	13.0
75%	19.0
max	57.0



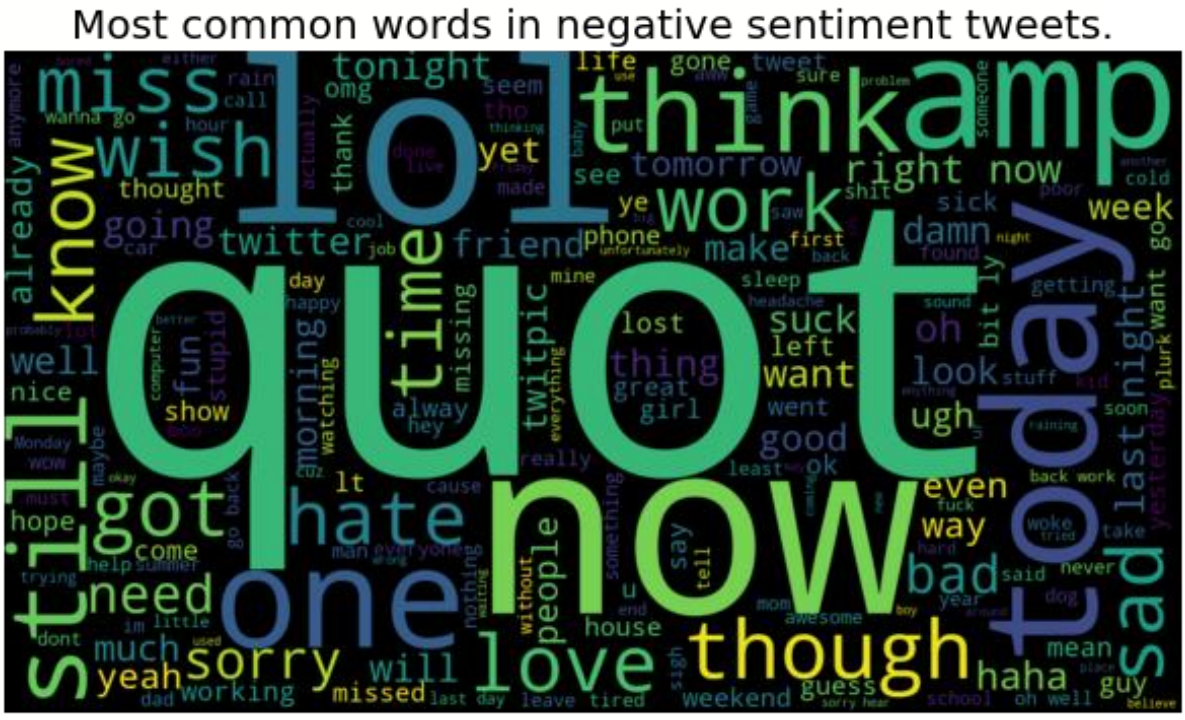
Now lets explore the most common words in tweets:

1. Positive class



Based on the word cloud, it can be deduced that the most repeated words in tweets with positive sentiment are words such as: love, quot, lol, amp,thank, haha.

2. Negative class



Based on the word cloud, we observed that the most repeated words in tweets with negative sentiment are words such as: quot, lol,today, amp which are the same as for positive sentiment class.

However, there are also word occurrences from which negative sentiment of a tweet can be inferred such as: miss, sorry,sad, hate,now etc.

 # DATA PREPROCESSING:

Since the tweets contain various **#,emojis,emoticoins,short words,@,etc.** So we need to clean the text column to get rid of these unnecessary items to train our model.

We are doing the following procedure to clean the tweets:

- ☐ *replace handwritten emojis with their feeling associated*
- ☐ *convert to lowercase*
- ☐ *replace short forms used in english with their actual words*
- ☐ *replace unicode emojis with their feeling associated*
- ☐ *remove emojis other than smiley emojis*
- ☐ *remove NON- ASCII characters*
- ☐ *remove numbers # re.sub("\d+", "", t)*
- ☐ *remove '#'*
- ☐ *remove '@'*
- ☐ *remove usernames*
- ☐ *remove retweet 'RT'*
- ☐ *remove links (URLs/ links)*
- ☐ *remove punctuations*
- ☐ *removes single letter words*

	polarity	text	text_cleaned
541200	0	@chrishasboobs AHHH I HOPE YOUR OK!!!	ahhh hope your ok
750	0	@misstoriblack cool , i have no tweet apps for my razr 2	cool have no tweet apps for my razr
766711	0	@TiannaChaos i know just family drama. its lame.hey next time u hang out with kim n u guys like have a sleepover or whatever, ill call u	know just family drama its lamehey next time hang out with kim guys like have sleepover or whatever ill call
285055	0	School email won't open and I have geography stuff on there to revise! *Stupid School* :'(	school email will not open and have geography stuff on there to revise stupid school sad
705995	0	upper airways problem	upper airways problem

The column 'text_cleaned' shows the cleaned tweets

Next, we applied stemming to the text_cleaned column to get the base form of each word.

And then we applied TFIDF vectorizer to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction.

After this step, we splitted the data into training and testing.

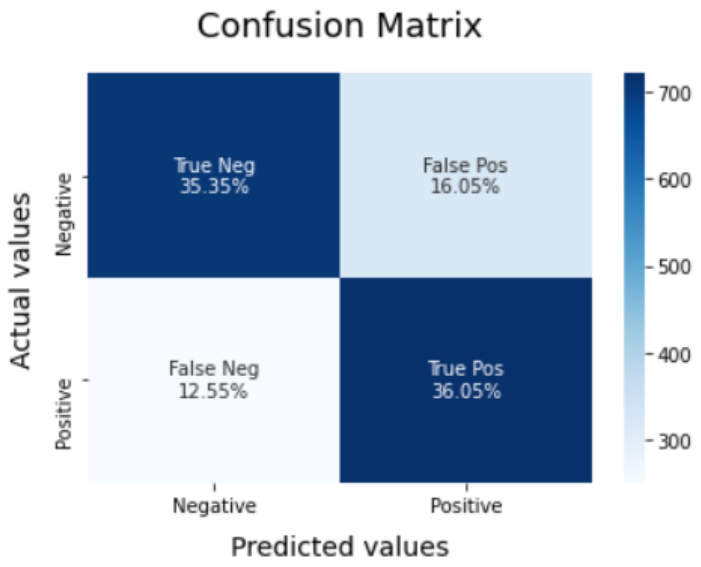
X_train (1280000, 368030)

```
y_train (1280000,)

X_test (320000, 368030)
y_test (320000,)

# MODEL EVALUATIONS:
```

Now it's time for training the models and compare their performance on various parameters.

1. LOGISTIC REGRESSION:

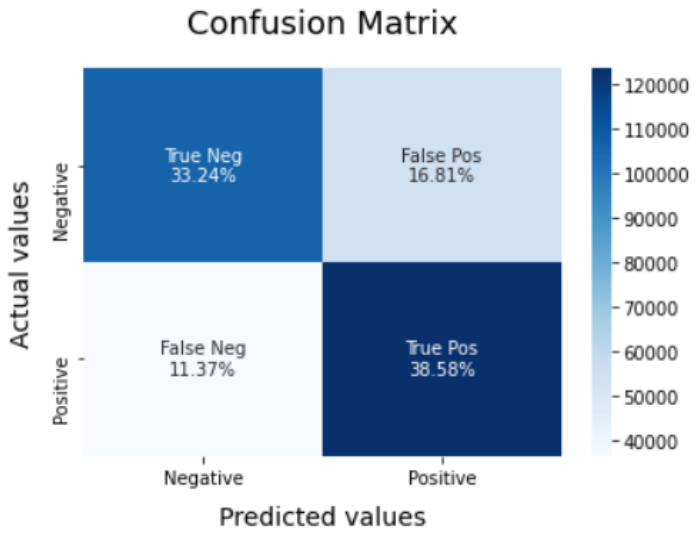
```
Accuracy of model on training data : 86.6625
Accuracy of model on testing data : 71.39999999999999

              precision    recall  f1-score   support

           0       0.74      0.69      0.71      1028
           1       0.69      0.74      0.72       972

    accuracy                           0.71      2000
   macro avg       0.71      0.71      0.71      2000
weighted avg       0.72      0.71      0.71      2000
```



2. RANDOM FORESTS:

```
Accuracy of model on training data : 73.5109375
Accuracy of model on testing data : 71.81875
```

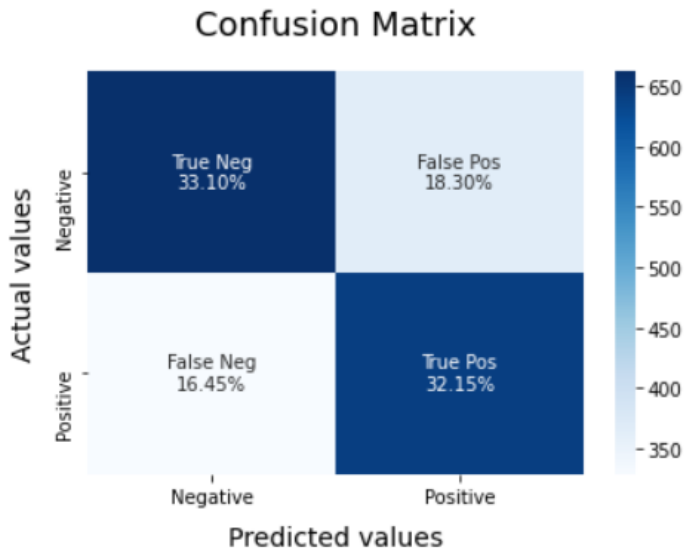
	precision	recall	f1-score	support
0	0.75	0.66	0.70	160150
1	0.70	0.77	0.73	159850
accuracy			0.72	320000
macro avg	0.72	0.72	0.72	320000
weighted avg	0.72	0.72	0.72	320000



3. MULTILAYER_PERCEPTRON_CLASSIFIER:

```
Accuracy of model on training data : 99.6875
Accuracy of model on testing data : 65.25
```

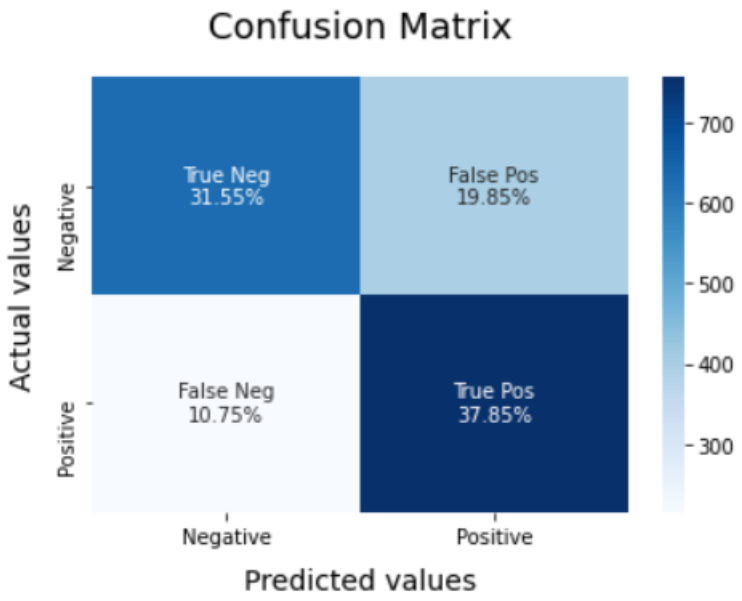
	precision	recall	f1-score	support
0	0.67	0.64	0.66	1028
1	0.64	0.66	0.65	972
accuracy			0.65	2000
macro avg	0.65	0.65	0.65	2000
weighted avg	0.65	0.65	0.65	2000



4. K-NEARESTNEIGHBOURS: using value of k as 90:

```
Accuracy of model on training data : 72.3875
Accuracy of model on testing data : 69.39999999999999
```

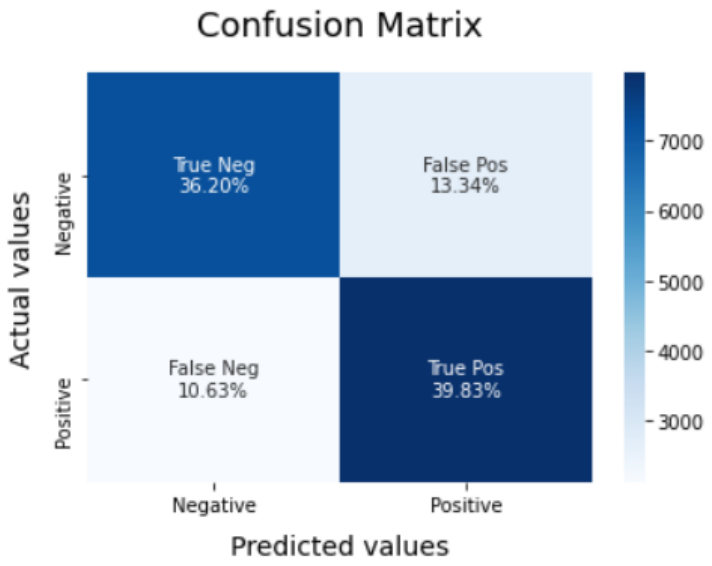
	precision	recall	f1-score	support
0	0.75	0.61	0.67	1028
1	0.66	0.78	0.71	972
accuracy			0.69	2000
macro avg	0.70	0.70	0.69	2000
weighted avg	0.70	0.69	0.69	2000



5. SVM linear:

```
Accuracy of model on training data : 84.23
Accuracy of model on testing data : 76.03
```

	precision	recall	f1-score	support
0	0.77	0.73	0.75	9909
1	0.75	0.79	0.77	10091
accuracy			0.76	20000
macro avg	0.76	0.76	0.76	20000
weighted avg	0.76	0.76	0.76	20000



So the ranked performance of each model found is:
SVM > RandomForest > LogisticRegression > KNN > MLP

CONTRIBUTION OF EACH TEAM MEMBER:

VIPUL : Exploratory data analysis
PIYUSH : Data preprocessing and data cleaning
P.SRUJAN : Model evaluations and comaprisons